

National University of Singapore
Department of Electrical and Computer Engineering

*LECTURE NOTES ON
EE 2002 ENGINEERING MATHEMATICS 2A*

MATLAB

Wang Qing-Guo

June 2001

INTRODUCTION TO MATLAB

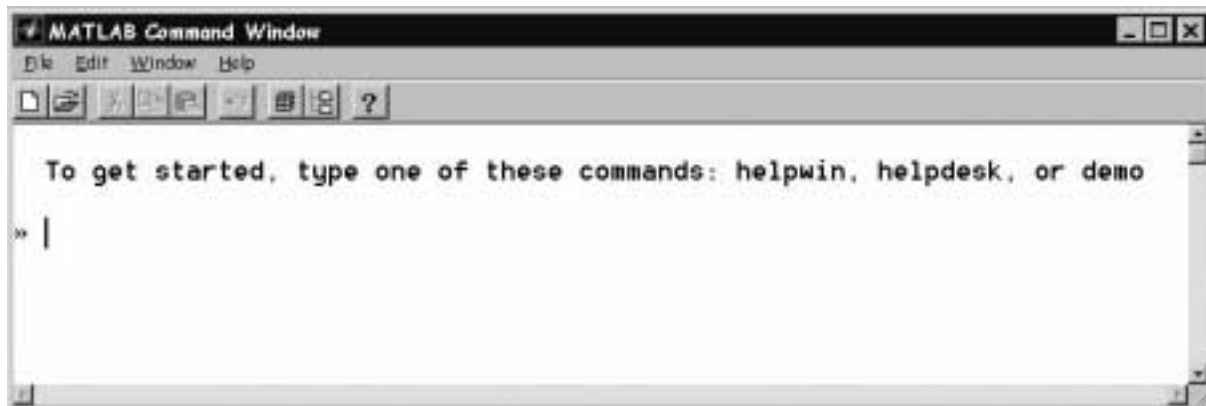
What is MATLAB ?

- MATLAB stands for MATrix LABoratory.
- Allows you to solve many technical computing problems, especially those with matrix and vector formulations, quickly.
- Typical uses include :
 - * Modelling, simulation and prototyping
 - * Data analysis and visualisation
 - * **Scientific and engineering graphics**
 - * Application development, including graphical user interface building.

Getting Started and Help Facility

Assume that you have created a shortcut for MATLAB or you are already in the directory where the MATLAB has been installed.

- Click the icon to start MATLAB or type “MATLAB”, followed by the **Return** key at the UNIX prompt.
- The MATLAB prompt “>>” should appear.



- Type “**helpdesk**” to access the Help desk and online manuals.
- Type “**help** *command*” to get online help.
- Type “**lookfor** *keyword*” to search for functions based

on a keyword.

- To quit MATLAB at any time, type **quit** or **exit** at the MATLAB prompt.

Data Entry and Arithmetic Expressions

Operations and commands in **MATLAB** are intended to be used in much the same way as writing a formula on paper. MATLAB works with essentially one kind of object: a rectangular numerical matrix that may possibly contain complex terms.

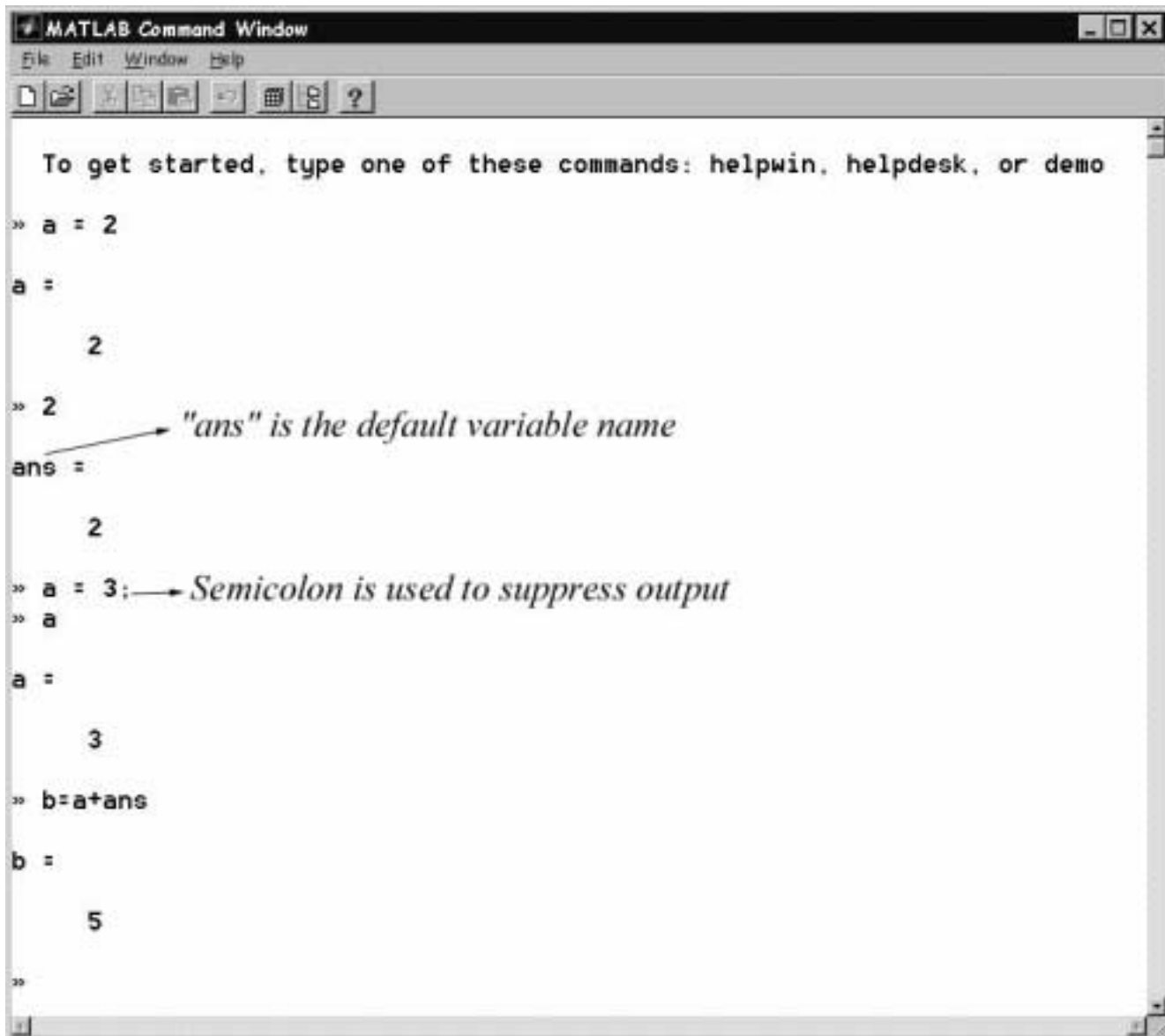
Scalar A one by one matrix.

Vector A matrix with one row or one column.

Operators MATLAB usually deals with vectors or matrices. There is a method of identifying arithmetic operations that act on a “member-by-member” basis rather than on the vector or matrix as a whole. The operator that specifies this action is the period (.) used in conjunction with one of the other standard MATLAB operators, such as * (multiply) and / (divide). For example, to multiply each component of a vector a by the corresponding components of a vector b having the same dimension, the appropriate operator is $a .* b$. Writing $a * b$ will generate an error message. Notice

the “space” after a and before the period (.). Although not always required, it is a good idea to insert this space because if instead of a we had a number, the “.” would be interpreted as a decimal point if there is no space. For example, try the command $3./a$ and $3.1a$ with a vector a .

Delimiter The semicolon (;) may be used to end a statement. if it is omitted, the result of executing the statement will be shown on the screen, and consequently, the delimiter may also be used to suppress display of MATLAB calculations.



The screenshot shows the MATLAB Command Window interface. At the top is a title bar 'MATLAB Command Window' with standard window controls. Below it is a menu bar with 'File', 'Edit', 'Window', and 'Help'. A toolbar with various icons is located below the menu bar. The main area contains the following text and commands:

```
To get started, type one of these commands: helpwin, helpdesk, or demo

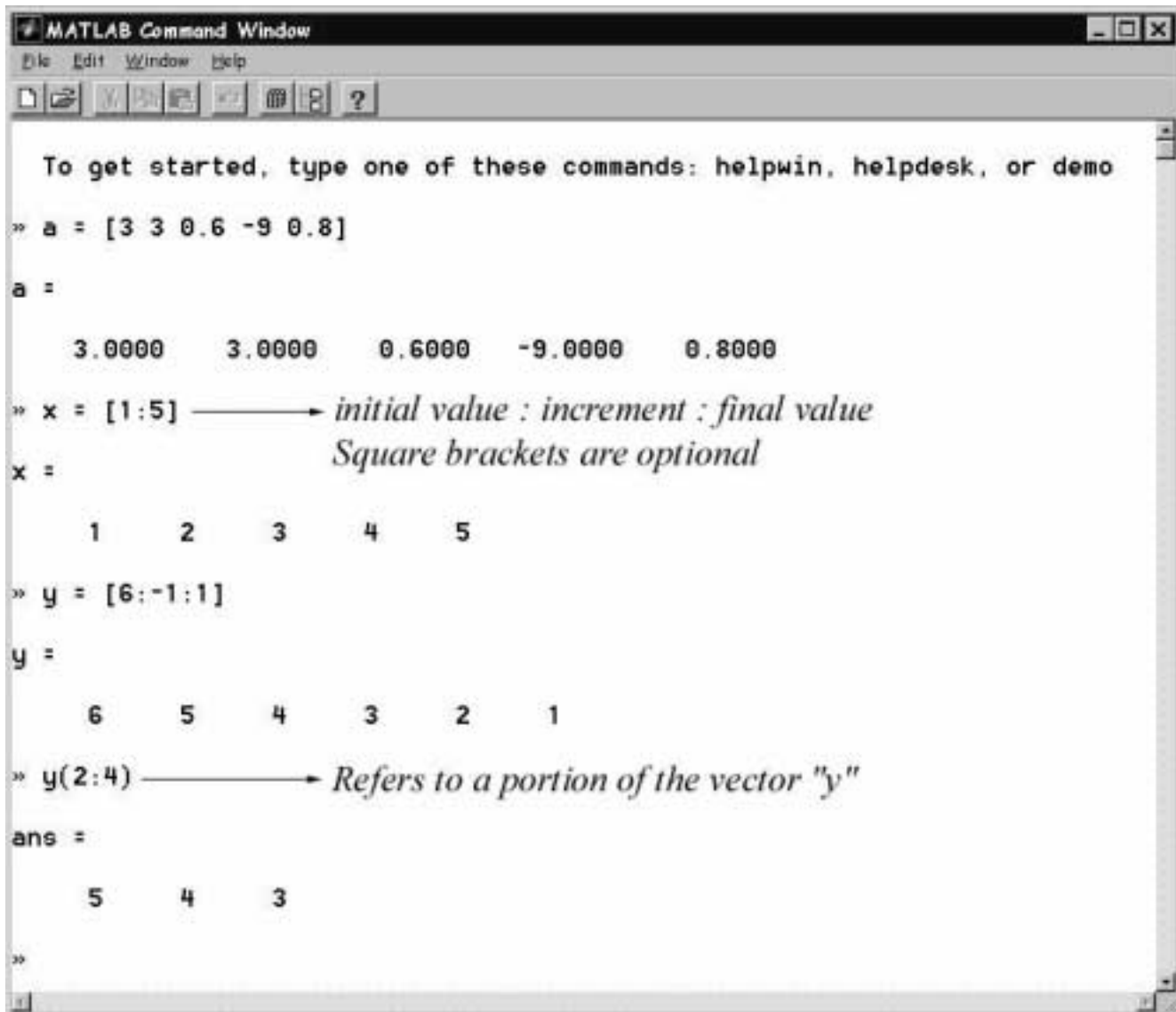
>> a = 2
a =
    2
>> 2
ans =
    2
>> a = 3;
>> a
a =
    3
>> b=a+ans
b =
    5
>>
```

Two annotations are present:

- An arrow points from the text *"ans" is the default variable name* to the output of the command `>> 2`.
- An arrow points from the text *Semicolon is used to suppress output* to the semicolon in the command `>> a = 3;`.

Vectors and the Colon Operator

Vectors are entered by putting numbers within square brackets (“[” and “]”).



The screenshot shows the MATLAB Command Window with the following content:

```

MATLAB Command Window
File Edit Window Help
To get started, type one of these commands: helpwin, helpdesk, or demo
>> a = [3 3 0.6 -9 0.8]
a =
    3.0000    3.0000    0.6000   -9.0000    0.8000
>> x = [1:5] —————→ initial value : increment : final value
                        Square brackets are optional
x =
     1     2     3     4     5
>> y = [6:-1:1]
y =
     6     5     4     3     2     1
>> y(2:4) —————→ Refers to a portion of the vector "y"
ans =
     5     4     3
>>

```


Workspace

Workspace Place where data are stored. if you type $A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$ and then [return] or [enter], the result displayed will be

```
A =  
    1    2    3  
    4    5    6  
    7    8    9
```

If you wish to access a particular component of a matrix, this can be done by specifying the appropriate row and column indices, as in

```
>> A(2,3)
```

which in this case yields the display

```
ans=6.
```

You can also access an entire row or column, for example, $A(3:)$ displays the third row of **A**. *Note that in MATLAB row and column indices always begin with number 1.* The

matrix **A** is *saved* for your later use *as long as MATLAB is running*. You could go on to add additional matrices and vectors, which would also be saved for later use.

who Used to see what is currently stored in the workspace with a listing of the names of the variables stored.

clear Used to clear the entire workspace.

clear x Removes just the variable **x** from the workspace.

save <filename> Workspace is saved in a file (no file extension is needed for DOS machines — the extension **.mat** is automatically assigned). To recall the workspace, use the **load** command.

load <filename> Again no extension is needed as MATLAB will automatically look for your <filename> among the files with **.mat** extensions. For Macintosh machines, MATLAB can be restarted by simply double-clicking on <filename>.

The screenshot shows the MATLAB Command Window interface. It displays a sequence of commands and their outputs, with handwritten annotations explaining the purpose of each command.

Initial State: The window shows a prompt and instructions: "To get started, type one of these commands: helpwin, helpdesk, or demo".

Commands and Annotations:

- `a = 10;`
- `b = [0:5:20];`
- `who` → *List current variables*

Output: "Your variables are:"

Name	Size	Bytes	Class
a	1x1	8	double array
b	1x5	40	double array

Summary: "Grand total is 6 elements using 48 bytes"

Next Commands and Annotations:

- `save temp` → *Save current variables in "temp.mat"*
- `clear` → *Delete current variables*
- `who`
- `load temp` → *Load data into workspace*
- `who`

Final Output: "Your variables are:" followed by the variable names "a" and "b".

Scripts and Functions

Modes of operation MATLAB commands are executed as they are entered into the keyboard. This is useful when trying new commands, executing short sequences of operations, troubleshooting, or in taking a more microscopic or macroscopic view of computed results. MATLAB can also be run by typing the name of a function or a program consisting of a set of MATLAB commands.

Much of MATLAB's power is derived from its extensive set of functions and the capability for the user to add new functions. Some functions are intrinsic to main MATLAB program, and others are available from the library of external files called **M-files**. Whether a file is intrinsic or contained in an M-file is transparent to the user. In either case, the user can call, or invoke, the function directly from within MATLAB. Refer to the user's manual or utilize the help facility of MATLAB to get more information on specific functions.

Script file An M-file without arguments. Similar to a macro in computer programming where a list of statements

is inserted into a program and then executed as a group. When a **scrip** is invoked, MATLAB simply executes the commands found in the file rather than waiting for input from keyboard.

Function file An M-file with arguments. It allows arguments to be passed to the function, unlike scripts that have no calling arguments. Additionally, variables may be defined and manipulated inside the function file that are local to the function and do not operate globally on the workspace. Function files are useful for creating new MATLAB functions written in the MATLAB language.

Script

- contains a sequence of valid MATLAB commands.
- can be written and modified by any editor.
- May include reference to other M-files.
- “%” may be used to provide comments.
- is saved as a file with .m extension.
- is executed by typing the filename without extension when you are at MATLAB prompt.

```
% This is a M-file - example.m  
x = [1:5];  
average = sum(x)/5
```

Functions

- Extension of file is also the letter “m”.
- Functions can accept input arguments and return output values.
- First line must be of the form

`function output = function-name(input)`

- Comment lines following the line “`function`” are displayed when the command “`help function-name`” is issued.

```
function y = mean(x)
% MEAN Average or mean value
% For vectors, mean(x) returns the mean ...
% For matrices, mean(x) is a row vector ...
% For N-D arrays, mean(x) is the mean ...
```

Element-by-element Arithmetic Operations

- Element-by-element arithmetic operators are : $.^*$, $./$, $.^$

- EXAMPLE

Evaluate $y(x) = x^2 e^{-x^2} + \sin(x) + 1$

when $x = -1, -0.999, -0.998, \dots, 0.999, 1$

* Solution 1

```
% prog1.m
clear;
x = [-1:0.001:1];
START = clock;
for i = 1:2001
    y(i) = x(i)^2*exp(-x(i)^2)+sin(x(i))+1;
end
etime(clock,START)
```


* Solution 2

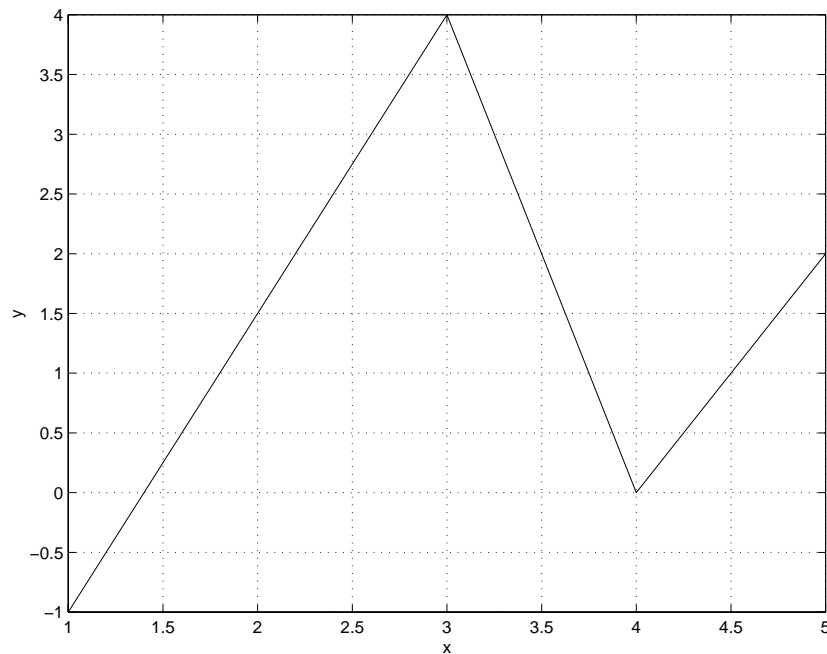
```
% prog2.m  
clear;  
x = [-1:0.001:1];  
START = clock;  
y = x.^2.*exp(-x.^2)+sin(x)+1;  
etime(clock,START)
```



Plotting

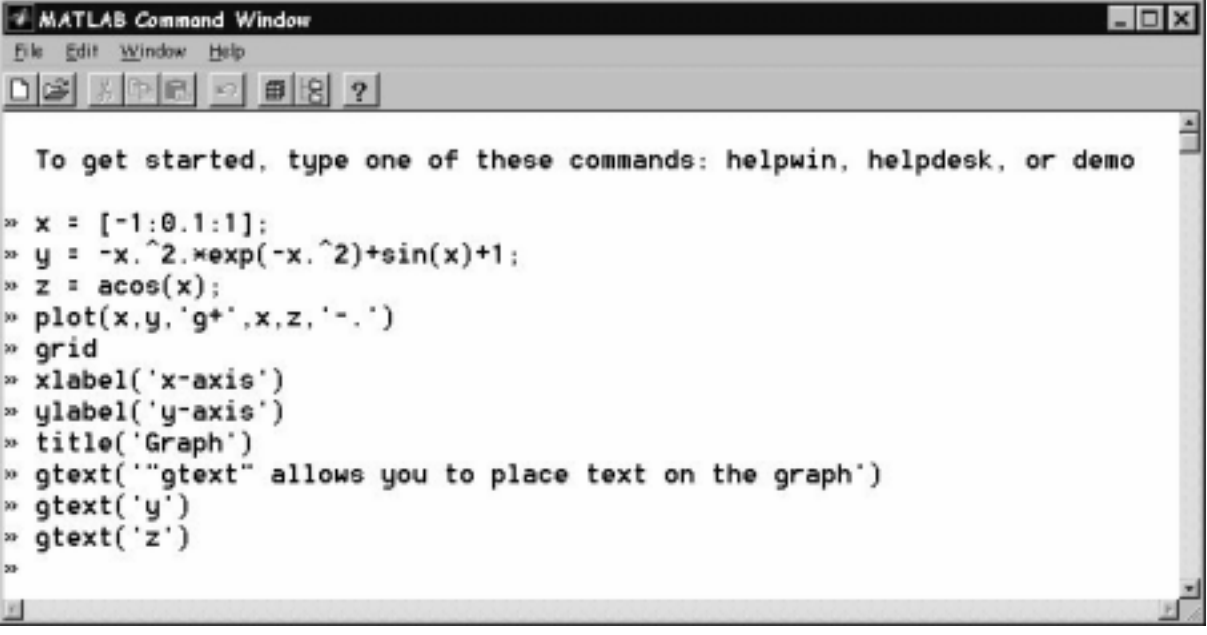
One of the principal attractions of MATLAB is its integrated plotting and graphics capability. There are various ways to plot data into the screen.

- “`plot(X,Y)`” plots vector **Y** verses vector **X** e.g. $X = [1 \ 3 \ 4 \ 5]$ and $Y = [-1 \ 4 \ 0 \ 2]$

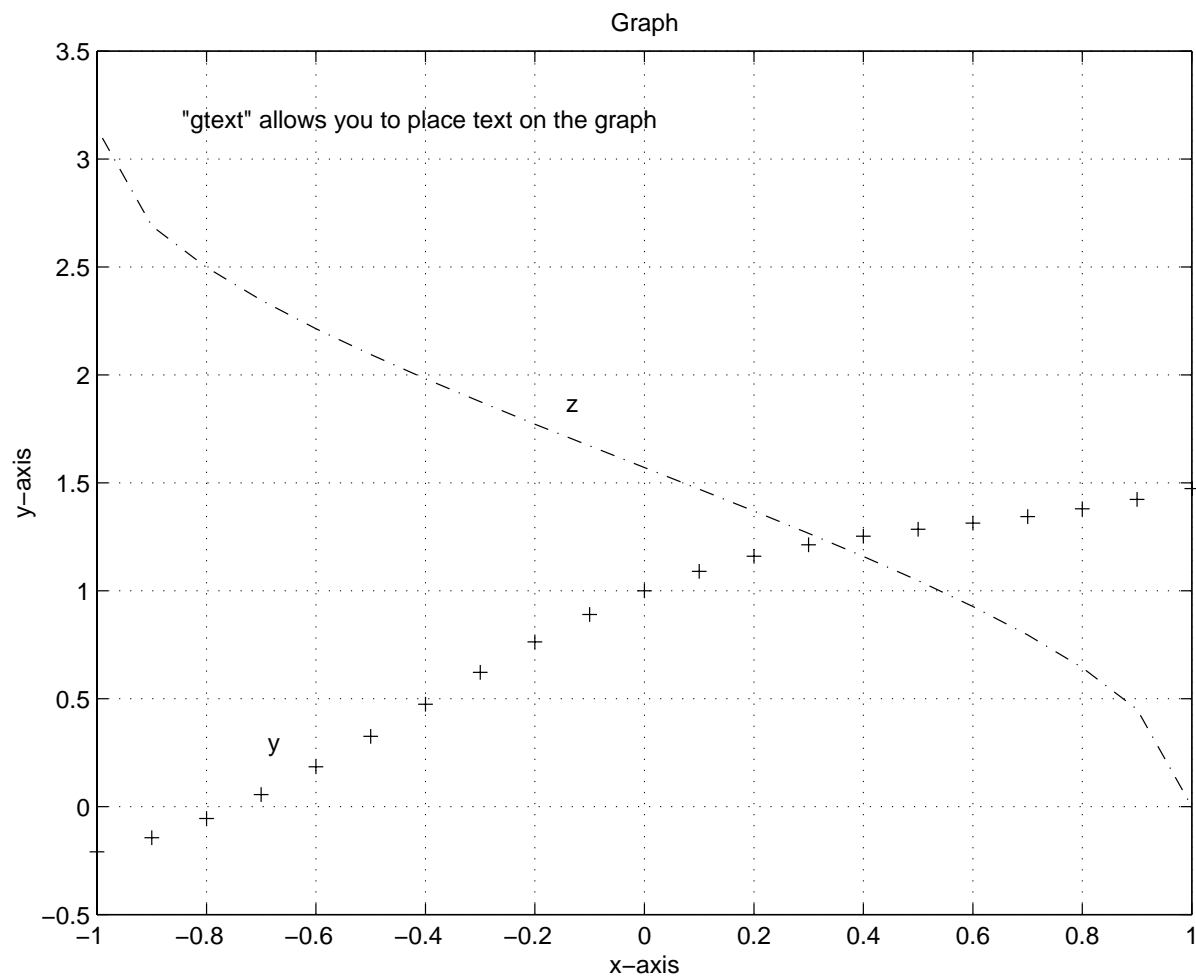


- The line style can be specified using the command “`plot(X,Y,S)`” where **S** is a 1,2 or 3 character string

<i>colour</i>		<i>symbol</i>	<i>line</i>
y yellow	g green	o circle	- solid
m magenta	b blue	x x-mark	: dotted
c cyan	w white	+ plus	-. dash-dot
r red	k black	* star	- - dashed

A screenshot of the MATLAB Command Window. The window has a title bar 'MATLAB Command Window' and a menu bar with 'File', 'Edit', 'Window', and 'Help'. Below the menu bar is a toolbar with icons for file operations and help. The main area contains a text prompt 'To get started, type one of these commands: helpwin, helpdesk, or demo' followed by a series of MATLAB commands entered at the prompt. The commands are: x = [-1:0.1:1]; y = -x.^2.*exp(-x.^2)+sin(x)+1; z = acos(x); plot(x,y,'g+',x,z,'-.-'); grid; xlabel('x-axis'); ylabel('y-axis'); title('Graph'); gtext('""gtext"" allows you to place text on the graph'); gtext('y'); gtext('z');

```
» x = [-1:0.1:1];
» y = -x.^2.*exp(-x.^2)+sin(x)+1;
» z = acos(x);
» plot(x,y,'g+',x,z,'-.-')
» grid
» xlabel('x-axis')
» ylabel('y-axis')
» title('Graph')
» gtext('""gtext"" allows you to place text on the graph')
» gtext('y')
» gtext('z')
»
```



References

- [1] M. Tummala, *Getting started with Matlab*, Naval Post-graduate School.